NAME: ______________________________

SIGNATURE: ________________________

STUDENT #: ________________________

UNIVERSITY OF SASKATCHEWAN
Department of Computational Science

CMPT 111.3
FINAL EXAMINATION
Dec 21, 1993

TOTAL MARKS = 100 TIME = 3 HOURS

Open Book

This exam consists of short answer questions and programming problems. **For each question**, write your answer legibly **on this question sheet**. You may use the backs of pages or answer books for rough work. The marks for each question are indicated in the margin. Allocate your time accordingly. Ensure that your **name** AND **student number** are written on the examination paper and that your name is on every page. When you have finished the exam, hand in the examination paper and ALL answer books to your instructor.

Marks

7 1) What is printed by the following Pascal program? If ANY value would be undefined at the time of printing, indicate this with a question mark, and assume the program continues without generating an error.

```
program Scope_question;
   var   A, B, C, D: integer;

procedure A1 (var B:integer; C:integer);
 var  D: string[10];
 begin
    writeln ('A1-start ',A, B, C, D);
   A := A + 1;
   B := A div 2;
   C := C + 1;
    D := 'FINAL';
    writeln ('A1-end ',A, B, C, D);
 end;

procedure A2;
 begin
    writeln ('A2-start ',A, B, C, D);
 end;

 begin   {main body}
   A := 3;
   B := 5;
   C := A + B;
   D := 0;
    writeln ('MAIN-start ',A, B, C, D);
   A1(B,C);
    writeln ('MAIN-end ',A, B, C, D);
   A1(C,B);
   A2;
 end.
```

5 2. a) Explain what the following mystery procedure does. The initial values sent in for parameters x and y are 1 and 100, respectively.

```
procedure foo (var A: arraytyp; x, y: integer);
    {assume that A is a 1-dimensional array containing 100 integer values}
    var  temp: integer;
begin
    if x <= y then begin
        temp := A[x];
        A[x] := A[y];
        A[y] := temp;
        foo(A, x + 1, y - 1)
    end
end;
```

5 b) Explain what the following mystery procedure does. The initial value sent in for parameter N is 100.

```
procedure foobar (var A: arraytyp; N: integer);
    {assume that A is a 1-dimensional array containing 100 integer values}
    var  i, temp: integer;
begin
    if N > 1 then begin
        foobar(A, N - 1);
        temp := A[N];
        for i := N downto 2 do
            A[i] := A[i - 1];
        A[1] := temp;
    end
end;
```

10 3. Santa Claus is planning to use a computer to help with gift distribution this year. He wants to build an array of records, each entry of which describes the toys to be left with a particular child. Prepare the type definitions and var declarations for this data structure, each record of which contains the fields shown below. Santa needs only to keep track of his favourite 1000 children.

Information required:
the child's name
the child's city
the child's age
an array for up to 25 gifts:
- description of gift (a string)
- cost (a real number)
- # of years of warranty (an integer)

Be sure to declare both the record and the array of records.

9 4. Given the strings S1, S2, and S3 below, answer each of the following questions:

```
S1 := 'santa claus is coming to town';
S2 := 'the big ho ho ho';
S3 := 'who loves ya baby'
```

a) What is the value of

```
pos (copy (S3, 2, 2), S2);
```

b) What is the value of

```
pos ((concat (copy (S2, 1, 1), copy (S3, 6, 1), copy (S3, 1, 1))), S1);
```

c) What is the value of

```
concat (copy (S2, 1, 4), copy (S3, 14, 4), copy (S3, 4, length ('yes')+4), copy (S1, 1, 5))
```

10 5. **Write a procedure** called "Stripit" with two parameters. The first parameter is a maximal length string (i.e. 255 characters) and the second is an integer. Your procedure is to remove all blanks from a string supplied as the argument, returning the reduced string through the same parameter and a count of the number of blanks removed through the integer parameter.

An example of how Stripit should work is as follows:

```
S:= 'The sky is so blue in Saskatchewan';
Stripit(S, N);
writeln(S);   →  writes the string   TheskyissoblueinSaskatchewan
writeln(N);   →  writes the number   6
```

12 6. The following procedure has been written to insert a node into its appropriate place in an ordered list using an iterative (non-recursive) approach. The sorting field is an integer identification number and the nodes are to be maintained in decreasing order. Some parts of the procedure have been left out. Complete the procedure by filling in the blanks appropriately.

```
type
   linktype = ^nodetype;
   nodetype = record
       value: integer;
       link: linktype
       end;
var
   LL, T: linktype;
   i: integer;

procedure insert (var Listptr: ___________; Nptr: ___________);
{ This procedure inserts a node pointed at by the parameter Nptr into an ordered linked list }
{ pointed at by the parameter Listptr. List is ordered in descending order of values. }
   var
       L1, L2: linktype;
       found: boolean;
begin
   if Listptr = nil then
       Listptr := ___________          { add to an empty list }
   else if Nptr^.value > ___________ then begin     { insert at front of list }
       Nptr^.link := ___________;
       Listptr := ___________
   end
   else begin  { begin a search for the proper position }
       L1 := Listptr;
       L2 := Listptr^.link;     {set pointers in preparation for search }
       found := ___________;
       while (L2 <> nil) and not found do begin     { find the insertion position }
           if Nptr^.value > L2^.value then
               found := ___________   { found it }
           else begin  { keep looking }
               L1 := ___________;
               L2 := ___________
           end;
       end; { end of while loop }
       Nptr^.link := ___________; {new node goes in between these two adjacent nodes}
       L1^.link := ___________;
   end
end;
```

8 7. Consider the following code:

```
type ptr = ^integer;  ptrptr = ^ptr;  ptrptrptr = ^ptrptr;
var P, Q: ptr;  R: ptrptr;  S: ptrptrptr;

new (P);  P^ := 111;
new (R);  R^ := P;
new (S);  S^  := R;
Q := P;
```

Four of the following have the value 111. **Circle** these four.

P P^ P^^ P^^^ Q Q^ Q^^ Q^^^ R R^ R^^ R^^^ S S^ S^^ S^^^

8. Suppose that a node type is declared as follows:

```
type
   pntr = ^recordtype;
   recordtype = record
                  data1, data2 : integer;
                  next : pntr
                end;
var
   LIST, ptr1, ptr2 : pntr;
   temprecord : recordtype;
```

Assume that you have a linked list of nodes with LIST pointing to the first node. The following syntactically correct code has been proposed to delete all nodes on the linked list:

```
        DISPOSE(LIST);
        LIST := NIL;
```

4 a) What is wrong with this approach?

10 b) Design and implement a **recursive** Pascal function "delete_all", which disposes of **all** the nodes in a link list.

20 9. Residents of Saskatoon pride themselves on their ability to survive more cold days than any other city in Canada. A prize, "The Cold Shoulder Award", donated by Lucien Bouchard, is to be awarded to English Canada's coldest city. The City of Winnipeg has challenged Saskatoon's claim as Canada's coldest city. Statistics Canada has hired you to settle this dispute. You have been instructed to find out which city (Saskatoon or Winnipeg) most often had the colder high temperature on the days between October 1, 1992 and March 31, 1993 (inclusive). Your task is to write a Pascal program that looks at Environment Canada's temperature database and determines whether Saskatoon or Winnipeg is colder. You will need to display the dates when Saskatoon has a colder high-temperature, display the dates when Winnipeg has a colder high temperature, display the dates when both cities have the same high-temperature, count the number of days each city is the colder than the other and declare a winner of the Cold Shoulder Award. Since Saskatoon and Winnipeg are the only serious contenders for this award, you need not process the data for other cities.

Environment Canada's temperature database contains a large amount of information. For each day of 1992 and 1993 all of Canada's major cities are listed, along with their daily high and low temperatures. The dates are in the form yymmdd (read as a longint). The cities are in no particular order, but for every date, both Winnipeg and Saskatoon have entries. The daily high and the daily low appear on the line in front of the city name. At the end of the list of cities for each day, there is a sentinel data value of -999. The data file looks like this:

data file: **environ.can**

```
920101
  10   - 8      Calgary
- 20   - 32     Winnipeg
- 10   - 15     Regina
- 18   -20      Saskatoon
   etc (for all the other cities)
. . .
-999
920102
- 14   - 22     Winnipeg
- 18   -22      Saskatoon
- 13   - 18     Regina
   etc (for all the other cities)
. . .
-999
. . .
```

etc. for each date until 931221 (today)

The general algorithm you should follow is given below.

```
For each day:
    check to see if the day is in the range Oct 1, 1992 to Mar 31, 1993
        if not, bypass all the information until the next date
        if so, see which city has the colder high-temperature that day
            Put the date into the appropriate Array (Saskatoon, Winnipeg or Tie)
            Update the appropriate counter.
            When you've found both cities' temperatures for the day, skip over the
              data for the other cities for that day
Figure out which city wins the Cold Shoulder award.
Print out each of the three sets of dates (dates when Saskatoon in colder, dates when
  Winnipeg is colder, and dates when both are equally cold)
Print out the counters and declare a winner (or a tie).
```